|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS 60 |
| Source:\* | Sherzod Elamanov (SyncTechno), Taehyun Kim (SyncTechno)  |
| Date:\* | 2023-06-26 |
| Reason for Change/s:\* | To control the update mechanism of a flexContainer resource type |
| CR against: Release\* | Release 5 |
| CR against: WI\* | [x]  Active WI-113 [ ]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [ ] mirror CR number: (Note to Rapporteur - use latest agreed revision)[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001 V5.2.0 |
| Clauses \* | 9.6.35, 10.2.4.18 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[ ]  Change to existing feature or functionality[x]  New feature or functionalityOnly ONE of the above shall be ticked |
| Other TS/TR(s) impacted | <TS/TR number>, <Version Number>, and <Description on which aspect should be reflected in this TS/TR> |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [x]  |
| Template Version: January 2020 (do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

If this is a correction, and the change applies to previous releases, a separate “mirror CR” should be posted at the same time as this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made e.g. a change impacting 5 tables should not only include a proposal to change only 3 tables. Include any changes to references, definitions, and abbreviations in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar.

Use change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the proposed new clause is located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to the content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

This CR proposes a new *updateMethod* attribute for <flexContainer> resource type. This attribute manages if Update operation for <flexContainer> should be updated partially or fully. It is useful in scenarios when an AE that is updating a <flexContainer> wants to delete optional customAttributes that are not present int the Update request from the old <flexContainer> representation.

TS-0004 has a solution for this problem, which is setting attributes that need to be deleted to NULL. However, this solution is not very efficient because if there are tens of customAttibutes that need to be deleted, the request content shall set all those customAttibutes to NULL. The solution in this CR targets this problem. With the proposed *updateMethod* attribute, the behaviour of flexConatiner can be configured with only one attribute.

### ----------------------Start of change 1-------------------------------------------

### 9.6.35 Resource Type *flexContainer*

The *<flexContainer>* resource type is a customizable container for data instances. It is a template for the definition of flexible specializations of data containers. Like a <*container*> resource, specializations of this *<flexContainer>* resource type are used to share information with other entities and potentially to track the data. While the <*container*> resources include data to be made accessible to oneM2M entities inside <*contentInstance*> children, a specialization of the *<flexContainer>* resource includes associated content directly inside the <*flexContainer*> by means of zero or more [*customAttribute*] attribute(s). The attribute name and attribute data type of [*customAttribute*] attributes are defined explicitly for each specialization of <*flexContainer>*, i.e. the specific set of attribute name and type are defined in a corresponding XSD-file.

Example usage of *<flexContainer>*: As a specialization of <*flexContainer*> that includes two [customAttribute] attributes, named "temperature" (xs:float type) and "humidity"(xs:positiveInteger type) can be specified in some TS. The actual data types of [customAttribute] will be described both in the specification document or XSD file which are referred by the value of *containerDefinition* attribute.

If a <*flexContainer>* resource is created with the *fcinEnabled* attribute present with value TRUE and at least one of the attributes *maxNrOfInstances, maxByteSize* or *maxInstanceAge* having non-zero values*,* then the hosting CSE shall automatically create a *copy* of the *<flexContainer>* resource in the form of a *<flexContainerInstance>* resource, child of the current *<flexContainer>* resource; and each time a <*flexContainer*> *custom* attribute is modified, a new <*flexContainerInstance>* child resource shall be added, which contains a copy of the *<flexContainer>* after update. In this case, virtual resources *<latest>* and *<oldest>* shall be available. The creation of <*flexContainerInstance*> ressources is controlled by the *fcinEnabled* attribute, following a retention policy specified by the *maxNrOfInstances, maxByteSize* and *maxInstanceAge* attributes.See clauses 10.2.4.16 and 10.2.4.18 for details.

When the *updateMethod* attribute is set to “All customAttributes” while updating a <*flexContainer*> resource, the Hosting CSE shall modify the [customAttribute] attributes of the *<flexConatiner>* resource present in the UPDATE request primitive and remove any [customAttribute] attributes from old *<flexContainer>* resource representation that are not present in the UPDATE request primitive. When the *updateMethod* attribute is set to “Modified customAttributes”, the Hosting CSE shall keep the not updated [customAttribute] attributes in the new <*flexContainer*> representation.

The *<flexContainer>* resource shall contain the child resource specified in table 9.6.35-1.

Table 9.6.35-1: Child resources of <*flexContainer*> resource

| **Child Resources of <*flexContainer*>** | **Child Resource Type** | **Multiplicity** | **Description** | *<flexContainerAnnc>* Child Resource Type |
| --- | --- | --- | --- | --- |
| *[variable]* | *<semanticDescriptor>* | 0..n | See clause 9.6.30 | *<semanticDescriptor>, <semanticDescriptorAnnc>* |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 | *<subscription>* |
| *[variable]* | *<container>* | 0..n | See clause 9.6.6 | *<container>**<containerAnnc>* |
| *[variable]* | *<flexContainer>* | 0..n | <flexContainer> resource can include any of its specializations as child resource | *<flexContainer>**<flexContainerAnnc>* |
| *[variable]* | *<flexContainerInstance>* | 0..n | Timestamped copy of the *<flexContainer>* resource | *<flexContainerInstance>* |
| *la* | *<latest>* | 0..1 | See clause 9.6.27 | *None* |
| *ol* | *<oldest>* | 0..1 | See clause 9.6.28 | *None* |
| *[variable]* | *<timeSeries>* | 0..n | See clause 9.6.36 | *<timeSeries>,**<timeSeriesAnnc>* |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 | *<transaction>* |
| *[variable]* | *<action>* | 0..n | See clause 9.6.61 | *<actionAnnc>* |

The *<flexContainer>* resource shall contain the attributes specified in table 9.6.35-2.

Table 9.6.35-2: Attributes of <*flexContainer*> resource

| **Attributes of *<flexContainer>*** | **Multiplicity** | **RW/****RO/****WO** | **Description** | ***<flexContainerAnnc>* Attributes** |
| --- | --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | WO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. | NA |
| *expirationTime* | 0..1 (note) | RW | See clause 9.6.1.3. | MA |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *labels* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *creationTime* | 0..1(note) | RO | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 0..1(note) | RO | See clause 9.6.1.3. | NA |
| *stateTag* | 1 | RO | See clause 9.6.1.3.This *stateTag* attribute value shall be incremented when a custom attribute of the flexContainer is modified. | NA |
| *announceTo* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announceSyncType* | 0..1 | RW | See clause 9.6.1.3. | MA |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | OA |
| *creator* | 0..1 | RO | See clause 9.6.1.3. | NA |
| *custodian* | 0..1 | RW | See clause 9.6.1.3. | NA |
| *location* | 0..1 | RW | See clause 9.6.1.3. | OA |
| *fcinEnabled* | 0..1 | RW | Controls the creation of <flexContainerInstance> child resources. Valid values are “TRUE” and “FALSE”.  | OA |
| *maxNrOfInstances* | 0..1 | RW | Maximum number of direct child *<flexContainerInstance>* resources in the *<flexContainer>* resource. | OA |
| *maxByteSize*  | 0..1 | RW | Maximum size in bytes of custom attributes that is allocated for the *<flexContainer>* resource for all direct child *<flexContainerInstance>* resources. | OA |
| *maxInstanceAge* | 0..1 | RW | Maximum age of a direct child *<flexContainerInstance>* resources in the *<flexContainer>* resource. The value is expressed in seconds. | OA |
| *currentNrOfInstances* | 0..1 | RO |  Current number of direct child *<flexContainerInstance>* resources in the *<flexContainer>* resource. It is limited by the maxNrOfInstances. The currentNrOfInstances attribute of the *<flexContainer>* resource shall be updated on successful creation or deletion of direct child *<flexContainerInstance>* resource of *<flexContainer>* resource. | OA |
| *currentByteSize*  | 0..1 | RO | Current size in bytes of custom attributes stored in all direct child *<flexContainerInstance>* resources of the *<flexContainer>* resource. It is limited by the maxByteSize. The currentByteSize attribute of the *<flexContainer>* resource shall be updated on successful creation or deletion of a direct child *<flexContainerInstance>* resource of *<flexContainer>* resource. | OA |
| *containerDefinition* | 1 | WO | This contains an identifier reference (URI) to the <*flexContainer*> schema definition which shall be used by the CSE to validate the syntax of the <*flexContainer*> resource.This URI may refer to one of the oneM2M <*flexContainer*> definitions specified in the following documents:* Generic Interworking [6]
* AllJoyn Interworking [7]
* Home Domain Information Model [8]

A list of oneM2M <*flexContainer*> definitions is also provided in clause 9.6.1.2.2 of oneM2M TS‑0004 [3].Other URI for other *<flexContainer>* definitions may be specified. | MA |
| *ontologyRef* | 0..1 | RW | A reference (URI) of the ontology used to represent the information that is stored in the present *<flexContainer>* resource. | OA |
| *contentSize* | 1 | RO | Sum of the size in bytes of all of the custom attributes. | NA |
| *nodeLink* | 0..1 | RW | The resource identifier of a <node> resource that stores the node specific information of the NoDN on which the interworked service represented by this <flexContainer> resource resides. | OA |
| *updateMethod* | 0..1 | RW | Controls the update operation of [*customAttribute*] attributes. Valid values are “All customAttributes” and “Modified customAttributes”. “All customAttributes” means all [*customAttribute*] attributes shall be updated even if some are not present the update Content. “Modified customAttributes” means only [*customAttribute*] attributes present in the update Content shall be updated. | OA |
| *[customAttribute]* | 0..n | RW | Specialization-specific attribute(s). Name and data type defined in each specialization of <*flexContainer>* resource. | OA |
| NOTE: When an instance of <*flexContainer*> is a child of a <*flexContainer*> resource, these attributes can be optional. Their presence is determined by the respective definition referred to by the *containerDefinition* attribute. |

When a new <*flexContainerInstance*> child resource is created, the attributes in Table 9.6.35-3 shall be copied from the parent <*flexContainer*> resource to the new <*flexContainerInstance*> resource.

Table 9.6.35-3: Copied attributes of the parent <*flexContainer*> resource

| Attributes of *<flexContainer>* | Description |
| --- | --- |
| *fcinEnabled* |  |
| *labels* |  |
| *stateTag* |  |
| *[customAttribute]* | All custom attributes are copied |

### ----------------------End of change 1-------------------------------------------

### ----------------------Start of change 2-------------------------------------------

#### 10.2.4.18 Update <*flexContainer*>

This procedure shall be used for updating the attributes and the actual data of a *<flexContainer>* resource.

Table 10.2.4.18-1: <*flexContainer*> UPDATE

|  |
| --- |
| ***<flexContainer>* UPDATE** |
| Information in Request message | All parameters defined in table 8.1.2-2 apply with the specific details for:***Content*:** attributes of the <*flexContainer*> resource as defined in clause 9.6.6 which need be updated |
| Processing at Originator before sending Request | According to clause 10.1.4 |
| Processing at Receiver | According to clause 10.1.4A child *<flexContainerInstance>* resource shall be created by the Hosting CSE if the *fcinEnabled* attribute is present in the <*flexContainer*> resource with value TRUE and at least one of the *maxNrOfInstances, maxByteSize* or *maxInstanceAge* attributes is present with a non-zero value and the request contains either no attribute at all or at least one custom attribute of the <*flexContainer*> or if it is adding or updating the *fcinEnabled* or *labels* attributes. All the attributes from the <flexContainer> specified in Table 9.6.35-3 are copied to the new <*flexContainerInstance*>.If at least one of *maxNrOfInstances, maxByteSize* or *maxInstanceAge* is created, modified or deleted in the *<flexContainer>* resource update request, then the set of <*flexContainerInstances*> children resources, the *currentNrOfInstances* and the *currentByteSize* attributes shall be updated accordingly. If at least one of *maxNrOfInstances, maxByteSize* or *maxInstanceAge* is set to 0, all <*flexContainerInstances*> children resources shall be deleted, and the *fcinEnabled*, *maxNrOfInstances, maxByteSize* and *maxInstanceAge attributes shall be deleted.*When the *updateMethod* attribute is set to “All customAttributes”, the Hosting CSE shall modify the [customAttribute] attributes of the *<flexConatiner>* resource present in the request primitive and remove any [customAttribute] attributes from old *<flexContainer>* resource representation that are not present in the request primitive. When the *updateMethod* attribute is set to “Modified customAttributes”, the Hosting CSE shall keep the not updated [customAttribute] attributes in the new <*flexContainer*> representation. |
| Information in Response message | According to clause 10.1.4 |
| Processing at Originator after receiving Response | According to clause 10.1.4 |
| Exceptions | According to clause 10.1.4 |

### ----------------------End of change 2-------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change? (Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.)
* Are multiple changes in this CR clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.?